IN THE SPECIFICATION

Please replace the paragraph starting on page 4, line 5 with the following paragraph:

However, when its shift-in and shift-out signals are derived from a common frequency source or clock, so that there is a precise, known relationship between ~~them-~~ them - the FIFO is by definition ~~is~~ in the synchronous mode. The key advantages of the synchronous FIFO over the asynchronous FIFO, are as follows:

Please replace the paragraph starting on page 4, line 24 with the following paragraph:

Accordingly, the following discussion is focused on synchronous FIFOs. Typical FIFO includes a 64 Kbit depth and 80 bits width, whereas the word is typically 80 bits long. One pointer, read or write, points to only one word. However, the "read out" or "write in" operations can be implemented in 4 modes: (a) read out ~~80 -mode~~ 80-mode, (b) write in 80-mode (at each clock the whole word is read out or written in), (c) read out 40-mode, and (d) write in 40-mode (at each of two clocks the whole word is read out or written in).

Please replace the paragraph starting on page 5, line 12 with the following paragraph:

To prevent a "too fast" read out, a read ~~enables~~ enable clock is activated only after a first word is written into the FIFO memory buffer. Thus, the read out operation has to wait depending on the particular FIFO device between (2-4) clocks before the first word is read out. For the second word there is no such ~~a~~ limitation. Each time when a FIFO is completely read out, an Empty flag is generated by a state machine. An enabling signal delays the generation of an Empty flag, that is the change of the Empty flag state from "zero" to "one is delayed, by a waiting period between (2-4) clocks needed for the first word to be read out. This prevents ~~the~~ reading out the word that has not been written in yet. ~~The reading~~ Reading out the word that has not been written in yet is called underflow. The time needed for the first word to come out from the FIFO buffer is called the fall through time.

Please replace the paragraph starting on page 7, line 12 with the following paragraph:

The problem with this prior art "adder approach" is that the triple input carry look ahead adder/comparator is ~~to~~ too slow because to do additions and comparisons the prior art adder circuitry requires 6 nsec.

Please replace the paragraph starting on page 9, line 24 with the following paragraph:

The apparatus of the present invention further comprises: (4) a latch having a first input receiving the first set_output signal, and a second input receiving the synchronized output signal. A first latch_output presents the first set_output signal as a first latch_output signal, whereas a second latch_output presents the synchronized output signal as a second latch_output signal. The latch is configured to hold the first latch_output signal and the second latch_output signal until the first set_output signal and second set_output signal change logic states. The first latch_output signal represents an Almost Full output flag that is at a first logic state when a FIFO memory block is Almost Full, and is at a second logic state when the FIFO is Not Almost Full. The second latch output signal represents a Not Almost Full output flag that is at the first logic state when the FIFO is Not Almost Full and is at the second logic state when the FIFO is Almost Full.

Please replace the paragraph starting on page 10, line 22 with the following paragraph:

In one embodiment ~~fo~~ of the present invention, the synchronizer further includes an SR latch coupled to the second set state machine. The SR latch is configured to receive the second set_output signal from the second set state machine, and the reset signal from the second logic block, and is configured to time an output of the second set_output signal depending on the reset signal. A Flip-Flop (FF) block coupled to the SR latch is configured to receive the second set_output signal, and configured to time the presentation of the second set_output signal as the synchronized signal depending on an external timing signal.

Please replace the paragraph starting on page 18, line 11 with the following paragraph:

Referring still to FIG. 1, the apparatus 10 further includes a latch 33 having a first input 64 receiving the output signal 36 from the first set state machine 12, and a second input 70 receiving the synchronized output signal 50 from the synchronizer 52. A first latch output 42 presents the first set machine output signal 36 as a first latch output signal 44, and a second latch output 48 presents the synchronized output signal 50 as a second latch output signal 46. The latch 33 is configured to hold the first latch output signal and the second latch output signal until the first set machine output signal 36 and second set output signal 53 change logic states. The first latch output signal 44 represents an Almost Full output flag that is at a first logic state when a FIFO block is Almost Full, and is at a second logic state when the FIFO is Not Almost Full. The second latch output signal 46 represents the Not Almost Full output flag that is at the first logic state when the FIFO is Not Almost Full and is at the second logic state when the FIFO is Almost Full.

Please replace the paragraph starting on page 19, line 17 with the following paragraph:

Referring still to FIG. 1, in the preferred embodiment, the synchronizer 52 further includes an SR latch 54 coupled to the second set state machine 14. The SR latch 54 is configured to receive the second set output signal 53 from the second set state machine 14, and is configured to receive the reset signal 60 from the second set state machine logic block 30. The SR latch 54 is configured to time its output 57 of the second set output signal 53 depending on the reset signal 61. Please~~,~~ see discussion below.

Please replace the paragraph starting on page 22, line 23 with the following paragraph:

Thus, the present invention uses the first set state machine 12 and the second set state machine 14 that each handle two input clocks and a look-ahead signal. Each of the state machines 12 and 14 has four possible output states, as illustrated in TABLE 2. With design criteria requiring five input variables and eight output states, the implementation of two input set state machines 12 and 14 is far less complex than the implementation of a single state machine capable of handling all combinations. Furthermore, the simplicity of blocking logic blocks 28 and 30 allows one to maintain the simplicity of the set/set implementation of the present invention. While the use of smaller, more efficient set state machines 12 and 14 is superior to using a single more complex state machine, the present invention can be implemented using a single larger state machine without departing from the spirit of the present invention.

Please replace the paragraph starting on page 24, line 11 with the following paragraph:

Similarly, when the FIFO becomes Not Almost Full, i.e., the FIFO has more than (offset+ 1) words, the first set state machine 12 ~~(of FIG. 1)~~ (of FIG. 1) generates a ~~“set_output $_1$ ”~~ “set output $_1$” pulse (depicted as event diagram 180 of FIG. 4A) at the next rising edge of the first read clock (shown as event diagram 120 of FIG. 4A). The “set_output $_1$” pulse within the first read clock pulse 120 is then synchronized by a wCLK 240 which then sets the programmable Almost Full flag in the S-R latch and makes the programmable Almost Full go high (active low). The synchronizer block 52 consists of the SR-latch 54 followed by the flip-flop (FF) register 56 clocked by the wCLK clock 58. The SR-latch gets reset after the FF register 56 has been clocked high. The second set state machine is blocked/inhibited when the external flag is High, while the first set state machine is blocked/inhibited when the external flag is Low. This ensures that the final SR-latch (33 of FIG. 1) never receives simultaneous “set” and “reset” pulse which could lead into an illegal state for the SR-latch 33.